COURS DE DÉVELOPPEMENT WEB

Troisième partie: **Les feuilles de styles CSS**

Notions fondamentales

Table des matières

1. Introduction

Les feuilles de styles, également appelées **CSS** (**Cascading Style Sheets**), sont au web ce que les styles sont au traitement de texte. Associé à des balises html ou crées par des utilisateurs, les styles définissent la façon dont le navigateur présentera la page. Les styles permettent d'appliquer une charte graphique homogène tant au niveau des polices, des fonds, des bordures que des couleurs utilisées. Créer une feuille de style, c'est assurer une mise en pages cohérente et plus simple à mettre à jour. Ainsi, pour modifier l'apparence des titres d'un site de plusieurs pages, il est inutile d'ouvrir obligatoirement chaque page html pour appliquer la transformation.

Les astuces proposées dans ce cours vont permettre à mieux comprendre les avantages dans l'élaboration des mises en pages, Elle s'accompagne à vous aider de mieux agencer les éléments composants votre page.

Exemple :

HTML
(pas de CSS)

HTML + CSS

2. Syntaxe de base du CSS

***Principe** :*

Une règle de style comprend :

- ***Un sélecteur** : il s'agit des **balises** concernées par cette règle ;*

- ***Un bloc de déclarations*** *: il indique les* ***propriétés*** *à attribuer à ces balises.*

  Chaque déclaration est du type : ***propriété : valeur ;***

Voici la syntaxe générale d'une feuille de style CSS :

balise1 {

propriete1: ***valeur1***;

propriete2: ***valeur2***;

propriete3: ***valeur3***;

}

balise2 {

propriete1: valeur1;

propriete2: valeur2;

propriete3: valeur3;

propriete4: valeur4;

}

balise3 {

propriete1: valeur1;

}

Exemple

P

{

color: green;

font-size : 10px;

}

3. Emplacement

Vous avez trois possibilités :

- *Directement dans les balises du fichier HTML via un attribut style*
- *Dans l'entête **<head>** du fichier HTML*
- *Dans un fichier **.css***

1. ***Ecrire le code dans un élément HTML :***

Il s'agit d'écrire le code à l'intérieur de la page html au sein d'un élément STYLE. Le code ne pourra s'appliquer qu'à la page dans laquelle il est créé. Cette méthode n'est pas recommandée pour des raisons de maintenance d'un site web.

Exemple :

```
<!DOCTYPE >
<html>
  <head>
      <title>première exemple</title>
       <méta charset= "utf 8">
    </head>
      <body  Style=" background-color: yellow;">
         <h1>université virtuelle du sénégal </h1>
          <p   Style="color: blue; font-size: 10px;">bonjour UVS</p>
     </body>
</html>
```

2. **Ecrire le code CSS dans l'en-tête de la page html (feuille de style interne):**

```
<! DOCTYPE >
  <head>
      <title>première exemple</title>
       <méta charset= "utf 8">
       <Style type="text/css">
            P
                {
                color: green;
```

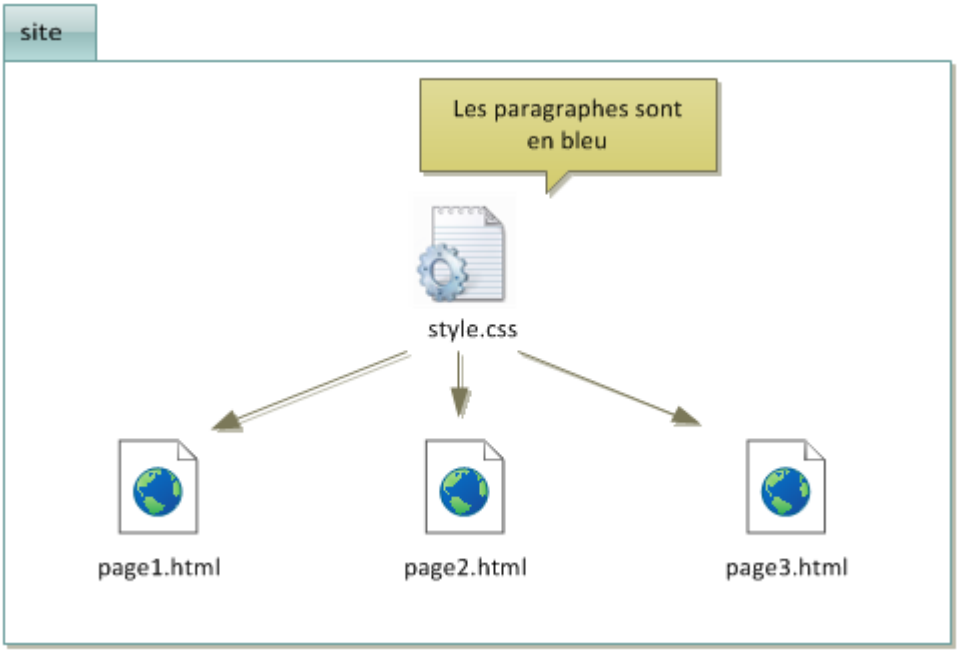
```
            font-size : 10px;
        }
    </Style>
</head>
    <body>
        <p>bonjour UVS</p>
    </body>
</html>
```

2. *Dans un fichier .css (méthode recommandée)*

Cette méthode est pratique parce qu'elle permet de séparer le code HTML de celui CSS : le fichier est plus facile à lire. C'est aussi la méthode qui est recommandée : si vous devez concevoir un site qui dispose de plusieurs dizaines de fichiers, la feuille de style CSS peut être utilisé par tous les fichiers du site web ; sinon (sans séparation HTML et CSS) le code CSS devra être copié dans les fichiers HTML qui utilise le code CSS.

Ainsi vous allez avoir deux fichiers : le fichier HTML et le fichier CSS.

NB : Si vous travaillez avec un fichier CSS externe, vous n'aurez besoin d'écrire cette instruction qu'une seule fois pour tout votre site, comme le montre la figure suivante.



Fichier HTML

```
<!DOCTYPE >
<html>
  <head>
        <title>première exemple</title>
        <méta charset= "utf 8">
        <link rel = "stylesheet" href="style.css" />
    </head>
      <body  Style=" background-color: yellow;">
          <h1>université virtuelle du sénégal </h1>
           <p>bonjour UVS</p>
    </body>
</html>
```

Remarque : Noter la nouvelle balise <link …/> dans l'entête du fichier HTML, elle indique le fichier est associé à un fichier nommé style.css.

Fichier CSS

```
p
{
color : blue ;
font-size:10px;
}
```

Remarque: Les fichiers HTML et CSS doivent se trouver dans le même dossier.

4. Sélecteurs et propriétés

Sélecteur & propriétés

4.1. Sélecteurs CSS simples

Un sélecteur est un des moyens les plus simples pour appliquer un style. Il permet d'affecter un style à un élément HTML. Dans une règle de style, le choix du sélecteur est extrêmement important : il indique les balises concernées par la mise en forme qui suit. Un sélecteur va donc nous permettre de cibler un ou plusieurs éléments HTML afin de leur appliquer un style particulier. Il existe différents types de sélecteurs : sélecteurs "***simples***" ou "***complexe***".

Un sélecteur ***simple*** permet de cibler les éléments qui correspondent au nom indiqué.

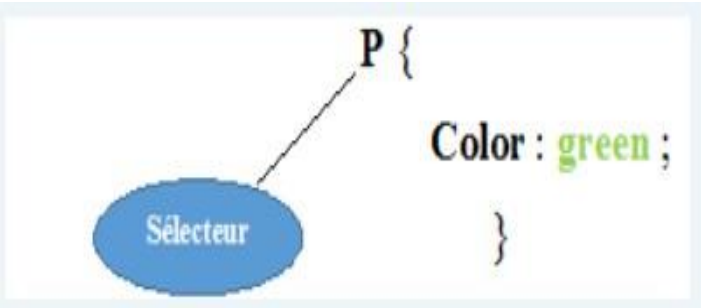
Exemple :

P {

color: green;

}

Dans cet exemple, le sélecteur ***P*** cible tous les paragraphes de notre page html.



4.2. Les propriétés CSS

Les ***propriétés*** nous permettent de choisir quel aspect ou quel style d'un élément la page html à modifier. Par exemple, nous allons pouvoir modifier la couleur d'un texte et lui appliquer la couleur que l'on souhaite grâce à la propriété ***color***

Exemple :

P {

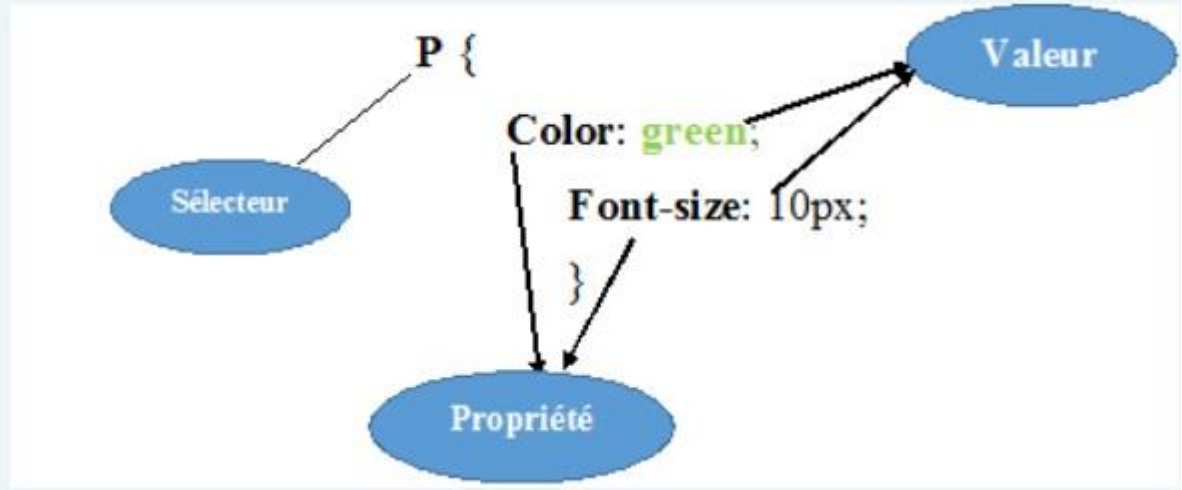
color : green;

font-size : 10px;

}

Dans cet exemple nous avons un *sélecteur* P avec deux propriétés (*color et font-size*).

Ici le sélecteur *P* cible tous les paragraphes de notre page html en mettant la couleur du texte en vert et la talle en 10.



5. Héritage en CSS

La notion d'héritage est très pratique en CSS. Elle signifie que tout élément html enfant va hériter "**en cascade**" des styles de ses parents. C'est de là que vient le nom du CSS: cascading stylesheets ou feuille de style en cascade.

Exemple : tous les éléments à l'intérieur de body sont des enfants de cet élément. Si on applique un style à l'élément body, ses enfants en hériteront automatiquement.

6. Les polices d'écriture en CSS

Le choix d'une police de caractères s'effectue à l'aide de la propriété ***font-family.*** Il permet de définir la police de notre texte.

Exemple :

On va définir une même couleur sur toute la police pour toute notre page HTML en appliquant notre propriété directement à l'élément ***body***.

```
<!DOCTYPE >
 <html>
   <head>
        <title>première exemple</title>
         <méta charset= "utf 8">
        <link rel = "stylesheet" href="style.css" />
     </head>
```

```
    <body >
        <h1>université virtuelle du sénégal </h1>
         <p>bonjour UVS</p>
    </body>
</html>
```

style CSS

```
body  {
    font-family:   Verdana;
     }
```

Remarque : il faut noter que tous les navigateurs selon la version possédée par le visiteur ne supportent pas la même police. Pour cette raison on indique plusieurs noms de police à utiliser en valeur de la propriété ***font-family, en*** commençant par celle souhaitée, et en séparant chaque valeur par une virgule.

Exemple :

```
body {
        font-family: "source code pro", verdana, sans serif;
    }
```

Lors de l'affichage sur cet exemple, le navigateur va d'abord tenter d'afficher la première police. Si celui-ci ne la supporte pas, il va prendre la seconde et etc.

6.1. Les propriétés de types font

Les propriétés CSS de type ***font-*** vont nous permettre de modifier l'apparence de notre police d'écriture, et donc de nos textes. Par exemple on peut transformer la taille, le poids, et le style de notre police.

Les propriétés de type font- les plus utilisées sont:

font-size: pour modifier la taille de nos textes ;

font-style: pour modifier le style de nos textes ;

font-weight: pour modifier le poids de nos textes.

La propriété de type font-size accepte deux types de valeurs :

- des valeurs de type absolu: souvent exprimées en pixel(**px**) ou en point(**pt**).

- les valeurs de type **relatif** exprimées soit en soit en **%** (pourcentage), en **em** ou en **ex.**

Exemple de **font-size:**

```
h1{
   font-size: 30px;
}
/*Notre paragraphe p aura une taille de 120%*/
 p{
   font-size: 150%;
}
h2 {
   font-size:  1.2em;

}
```

- La propriété font-style elle permet de forcer sur le style de la police. Elle accepte quatre valeurs différentes :
  - normal (par défaut) ;
  - italic (italique) ;
  - oblique (penché) ;
  - inherit (l'élément ciblé hérite du style de son élément parent). exemple:

```
P {          font-
style: italic;

      }
```

- Propriété de type font-weight:

Cette propriété peut prendre différentes valeurs :

- ***normal*** (valeur par défaut) ;
- ***lighter*** (la police sera plus fine) ;
- ***bold*** (la police sera plus épaisse) ;
- ***bolder*** (la police sera très épaisse) ;
- ***inherit*** (l'élément hérite du style de son parent) ;
- ***initial*** (définit la propriété sur sa valeur d'origine).

exemple:

```
/*Le texte de notre paragraphe p est épais (gras)*/

p{
   font-weight: bold;
}
/*Le texte de notre paragraphe p2 est normal*/
```

```
.p2{
   font-weight: 300;

}
```

6.2. Les propriétés de types Text

Les propriétés CSS de type **text** vont nous permettre de changer ***la mise en forme*** de nos textes et leur apparence.

Les propriétés les plus utilisées sont:

- La propriété **text-align** (gère l'alignement) ;
- La propriété **text-transform** (gère la mise en majuscules / minuscules) ;
- La propriété **text-decoration**(gère la décoration) ; La propriété **text-indent** (gère l'indentation) ; La propriété **text-shadow** (gère les ombres).

text-align:

valeurs possibles: left, right, justify, center.

exemple:

```
p {   text-align: center;
}
```

text-transform:

valeurs possibles:

- ***Lowercase*** : Met tout le texte en minuscules ;

- ***Uppercase*** : Met tout le texte en majuscules ;

- ***Capitalize*** : Met la première lettre de chaque mot en majuscule ;

- ***Inherit*** : Hérite de la valeur de l'élément parent ;

- ***None*** : Pas de transformation du texte. Utile pour annuler une transformation par défaut donnée par héritage.

Exemple :

```
p {
    text-transform: uppercase;
 }
```

text-decoration: Cette propriété accepte six valeurs différentes :

- ***Underline*** : Le texte sera souligné ;
- ***Overline*** : Une ligne apparaît au-dessus du texte ;

- ***Line-through*** : Le texte sera barré ;
- ***Inherit*** : Hérite de la valeur de l'élément parent ;
- ***Initial*** : Utilise la valeur par défaut de la propriété ;
- ***None*** : Pas de décoration.

Exemple:

```
p {
text-decoration: underline;
}
```

6.3. Les couleurs et l'opacité

La propriété ***color*** accepte différents types de valeurs:

- un nom de couleur:

Exemple : H1{

color: white;

}

La façon la plus simple de changer la couleur du texte est de donner son nom en plus sa valeur.

- une valeur hexadécimale : **#f5f5dc, #ffe4ba** ...

7. Les éléments HTML / CSS et les blocs

Element

7.1. ID et Class

Soit le code CSS suivant :

```
p {
    color: blue;
    font-size: 10px;
}
```

Tous les paragraphes qui utiliseront que le code précédent seront colorés en bleu et le texte aura la taille 10px.

Pour que certains paragraphes soient écrits de manière différente, on utilise les attributs class ou id.

L'attribut class aura pour valeur le **nom** qui sert à **identifier** la balise. Il peut être utilisé sur n'importe quelle balise.

<h1 class=" "> </h1>

Exemple:

Fichier HTML

```
<!DOCTYPE >
<html>
  <head>
      <title>première exemple</title>
       <méta charset= "utf 8">
      <link rel = "stylesheet" href="style.css" />
    </head>
      <body  Style="background-color: yellow;">
         <h1>université du Sénégal </h1>
          <p class="salutations">bonjour Monde</p>
          <p> Bienvenue sur la page dédiée aux informations sur les cours </p>
     </body>
</html>
```

Fichier CSS

```
.salutations
      {
          color: red;
      }
```

Dans cet exemple, tous les paragraphes de la page HTML qui ont comme nom *salutations* seront affichés en rouge.

L'attribut id fonctionne de la même manière que class mais il ne peut être utilisé qu'une seule fois dans le code. On mettra des id que sur des éléments qui sont uniques dans la page, le logo par exemple.

Dans le CSS, ce n'est pas un point que vous mettez avant le nom de l'id, mais un dièse (#) :

Syntaxe :

```
# nom-de-l'id
        {
          /* Mettez les propriétés CSS ici comme pour class*/
        }
```

7.2. Block et inline

En HTML, tout élément est soit de type block, soit de type inline.

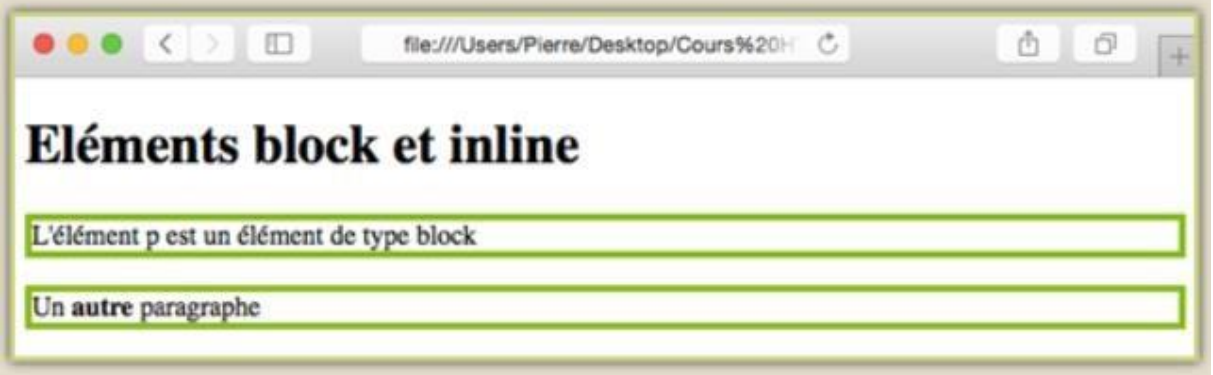
Un élément de type block occupe toute la largeur disponible dans la page. La balise <p> est un exemple typique d'élément block. Un élément de type block va toujours commencer sur une nouvelle ligne. Voici quelques éléments HTML de type block : h1,h2, ol, ul, form, div, …

Exemple

```html
<!DOCTYPE html>
<html>
    <head>
        <title>ELéments block vs inline</title>
        <meta charset= "utf-8">
        <link rel="stylesheet" href="styles.css">
    </head>

    <body>
        <h1>Eléments block et inline</h1>
        <p>L'élément p est un élément de type block</p>
        <p>Un <strong>autre</strong> paragraphe</p>
    </body>
</html>
```

```css
p{
    border: 4px solid #88BB11;
}
```



Dans cet exemple, une bordure a été créée pour les paragraphes. On remarque bien que la bordure prend toute la largeur de la page.

Un élément de type inline n'occupe que la largeur indispensable à l'affichage du contenu ciblé. Il n'entraine pas automatiquement un retour à la ligne.

Exemple

La balise est un exemple typique d'élément inline. Un élément de type inline va toujours commencer sur une nouvelle ligne. Voici quelques éléments HTML de type block: em, a, img, span, …

NB : Si les éléments de type block sont généralement ceux que l'on utilise pour la mise en page, les éléments inline sont surtout utilisés pour attribuer un style à une partie du texte.

7.3. DIV et Span

Soit le code HTML suivant :

** Bienvenue à l'UVS ! **

On veut mettre en bleu uniquement l'acronyme UVS. La balise class ne permet pas de le faire car c'est un attribut et il agit sur toute la balise. Dans ce cas, nous allons utiliser des balises comme ou .

La balise est une balise de type inline, elle se place au sein d'un paragraphe de texte pour sélectionner certains mots. La balise , quant à elle, est de type block et peut être utilisée pour entourer un bloc de texte.

Exemple

Pour mettre en bleu UVS, on utilise la balise :

Code HTML

```
<p>Bienvenue à l'<span class="universite">UVS</span>!</p>
```

Code CSS

```
.universite
        {
                color :blue ;
        }
```

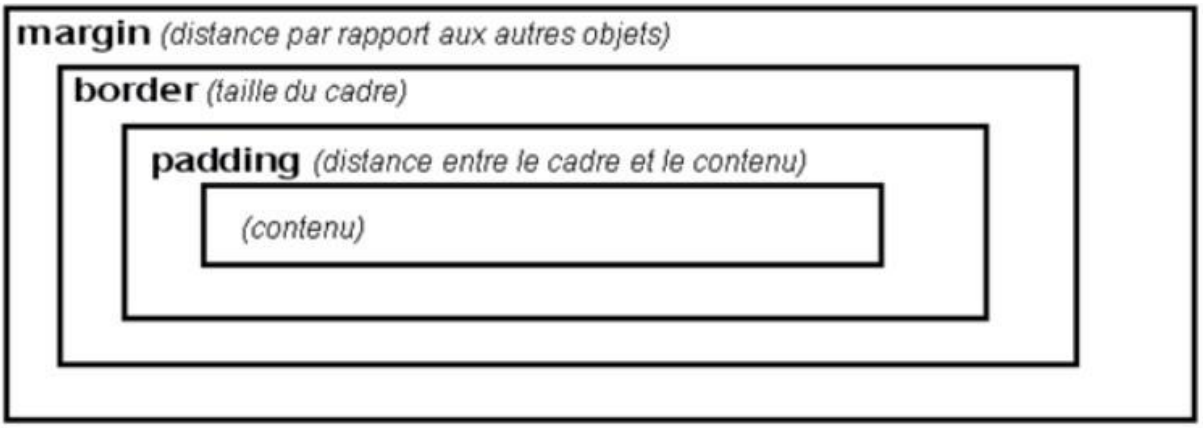
7.4. Les boites en CSS

Une page HTML peut être vue comme une succession et un empilement de boîtes. Dans une page web, tout élément HTML peut être considérée comme une boîte rectangulaire.

Les boîtes vous permettront de :

- définir les marges ***intérieures*** et ***extérieures*** et les ***bordures*** autour du contenu de l'élément ; -

positionner les éléments les uns par rapport aux autres de manière efficace.



Voici un tableau qui résume l'ensemble des attributs relatifs aux marges et bordures.

Attributs	Valeurs	se charge de	exemple
margin	pt, px, cm, %	4 marges	*body {margin:1cm;}*
margin-top		marge en haut	*p {margin-top:10px;}*
margin-bottom		marge en bas	*h3 {margin-bottom:3pt;}*
margin-left		marge à gauche	*img {margin-left:50px;}*
margin-right		marge à droite	*p.citation {margin-right:10pt;}*
border	pt,px, cm, %	largeur du cadre	*p {border:5px;}*
border-top			*h1 {border-top:0.2cm;}*
etc ...			
border-style		style de cadre	
	solid	ligne simple	p {border-style:solid;}
	double	ligne double	h1 {border-style:double;}
padding	pt,px,cm,%,etc	marge intérieures	p {padding: 5px;}
color	valeur hexa /nom	couleur d'un élément	#menu {color:#000000;} body {color:blue;}
background	aussi	couleur de l'arrière-plan	h1, h2 {background:silver;}

Source : http://tecfa.unige.ch/

7.5. Les bordures

De nombreuses propriétés CSS permettent de modifier l'apparence des bordures. border regroupe l'ensemble de ces propriétés ; on peut utiliser jusqu'à trois valeurs pour modifier l'apparence : largeur(en pixels), couleur (mettre le nom de la couleur ou la valeur hexadécimal ou valeur RGB), type de la bordure(none, solid, dashed, double,inset, ….).

Exemple : pour avoir une bordure rouge, en trait simple et épaisseur égale 3pixels, voici le code :

```
h1
{     border : 3px red solid
;
}
```

Vous avez la possibilité de mettre une bordure différente sur chaque côté : border-top, border-bottom, borderleft, border-right.

Exemple :

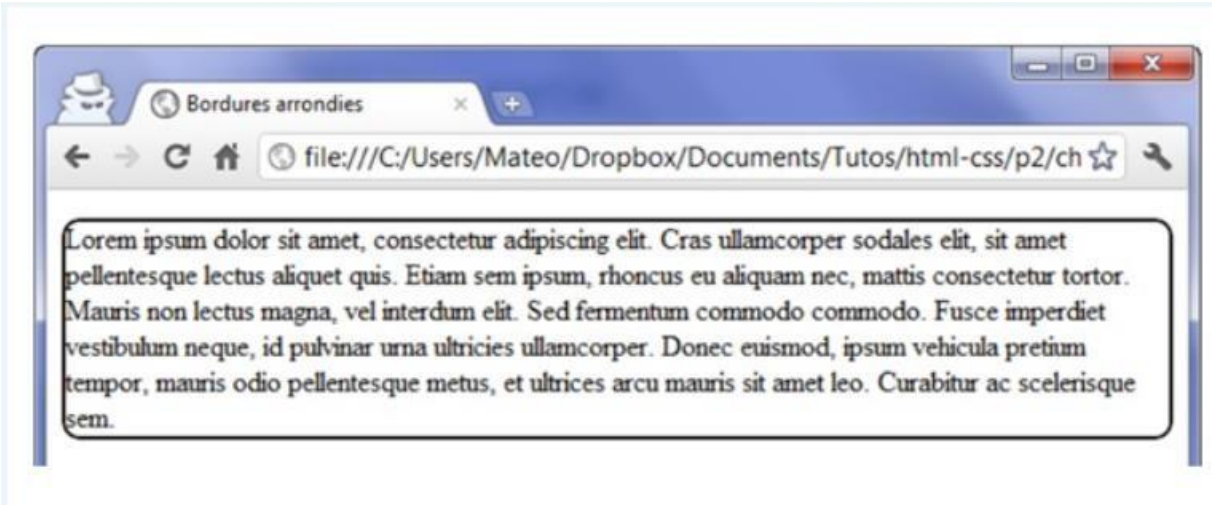
```
p {
        border-left : 3px red solid ;
        border-right :  3px blue solid ;
}
```

Vous pouvez arrondir les angles de la bordure en utilisant la propriété border-radius.

Exemple :

```
p{
        border-radius : 10px ;
}
```

Voici le résultat :



Le type de bordure : là, vous avez le choix. Votre bordure peut être un simple trait, ou des pointillés, ou encore des tirets, etc. Voici les différentes valeurs disponibles :

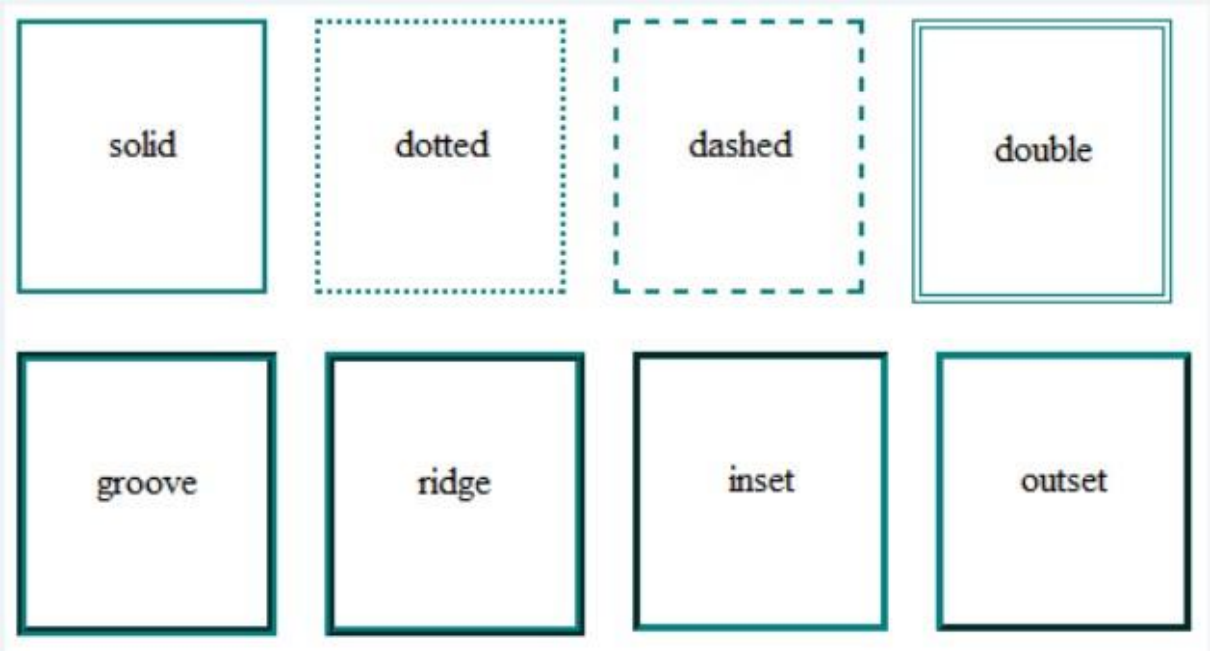
- **none** : pas de bordure (par défaut) ;
- **solid** : un trait simple ;
- **dotted** : pointillés ;
- **dashed** : tirets ;

- **double** : bordure double ;
- **groove** : en relief ;
- **ridge** : autre effet relief ;
- **inset** : effet 3D global enfoncé ;
- **outset** : effet 3D global surélevé.

Ainsi, pour avoir une bordure bleue, en tirets, épaisse de 3 pixels autour de mes titres

```
h1
{
border: 3px blue dashed;
}
```

La figure suivante vous présente les différents styles de bordures que vous pouvez utiliser.



8. Gestion des espaces

En CSS on peut contrôler l'affichage des espaces en utilisant les propriétés ***line-height, letter-spacing, wordspacing.***

- La propriété ***line-height*** vous permet de choisir la distance ou l'écartement entre deux lignes de texte.

exemple:

```
p
  {
    line-height : 20px;
  }
```

- Les propriétés CSS ***letter-spacing*** et ***word-spacing*** vont nous permettre respectivement de définir l'espace entre les lettres et entre les mots dans un texte.

Exemple :

```
<!DOCTYPE html>
<html>
    <head>
        <title>Les espaces</title>
        <meta charset= "utf-8">
        <link rel="stylesheet" href="styles.css">
    </head>

    <body>
        <h1>Gestion des espaces</h1>

        <p class="p1">Un premier paragraphe</p>
        <p class="p2">Un autre paragraphe</p>
    </body>
</html>
```

source: http://pierre-giraud.com

code css:

```
.p1{
    letter-spacing: 2px;
    word-spacing: 10px;
}

/*Le texte ira de gauche vers la droite !*/
.p2{
    text-align: center;
    letter-spacing: -25px;
}
```

8.1. Largeur et hauteur

Les propriétés CSS width et height permettent de contrôler ***la largeur et la hauteur*** *du contenu* des éléments de type bloc et ceux remplacés.

- Pour fixer la largeur vous pouvez utiliser la propriété **width**

Exemple : on va définir la largeur d'un **div**

```
div {

    witdth = 200 px;

    border = 1px  solid green;

}
```

- La propriété **height** permet de fixer la hauteur

  Exemple *: on va définir la hauteur d'un* ***div***

```
div {

        height = 300 px;

        border = 1px solid green;

 }
```

Il faut noter que les propriétés witdh et height prennent les même valeurs absolue et relative (utile dans le cas d'un site responsive) ou la valeur **auto (**la hauteur sera calculée automatiquement**).**

- ***Utilisation avec un modèle de boites***

Pour régler efficacement la hauteur et la largeur d'un élément, il faut avant tout bien avoir compris le modèle des boîtes. Il faut définir la largeur et la hauteur ensuite viendront s'ajouter les marges et les bordures.

Exemple :

```
<!DOCTYPE html>
<html>
   <head>
      <title>Le modèle des boîtes</title>
      <meta charset= "utf-8">
   </head>

   <body>
      <div>
         <h1>université Iba Der Thiam de Thies</h1>
         <p class="para1">filière MIO.</p>
         <p class="para2">cours Développement web</p>
</div>
   </body>
</html>
```

code css:

```
div{
        background-color: #088;

        width: 500px;

}

h1{
```

```
width:300px;
background-color: white;
height: 100px;
border: 1px solid black;
}

.para1{
width:300px;
background-color: orange;
padding: 60px;
border: 5px solid #FFF;
}
```

8.2. Les marges

En CSS, Il est important de comprendre comment sont calculées les dimensions des « boîtes », c'est-à-dire des blocs contenant texte et images. Les dimensions d'une boîte peuvent être fixées, ainsi que ses marges intérieures (à l'intérieur des bordures) et ses marges extérieures (à l'extérieur des bordures).

1. ***Marges intérieures :***

Les **marges intérieures** se trouvent entre le contenu de l'élément et sa bordure. Ainsi, définir une marge intérieure importante va éloigner la bordure de l'élément de son contenu. Si on définit une couleur de fond pour notre élément, celle-ci s'applique également dans l'espace correspondant aux marges intérieures.

Pour définir une marge intérieure, il faut utiliser la propriété *padding*- avec ses différentes positions :

top = en haut, bottom = en bas, left = à gauche, right = à droite exemple:

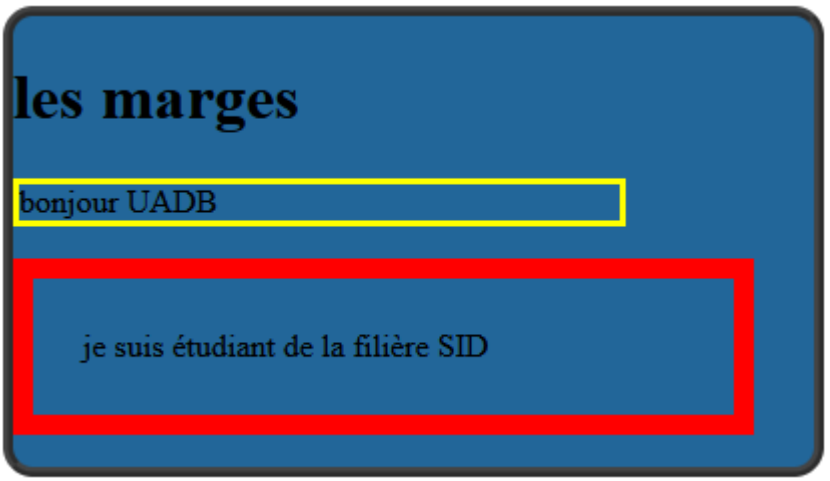
```
<!DOCTYPE html>
<html>
```

```
   <head>
      <title>modèle des boîtes</title>
      <meta charset= "utf-8">
     <link rel = "stylesheet" href="style.css" />
   </head>

   <body>
      <div>
         <h1>les marges</h1>
         <p>bonjour UVS</p>
         <p class="p1">je suis étudiant de la filière MIC</p>
</div>
   </body>
</html>
```

code css

```
div{
 background-color: #269;
width: 400px;
border: 5px ridge #444;
border-radius: 15px;
margin-top: 50px;
margin-left: 50px;
}
p {
   width: 300px;
   border: 3px solid yellow;
}
.p1{
         width: 300px;
         border: 10px solid red;
         padding: 25px;
}
```



Application 1 : Modifier le code précèdent pour reproduire le schéma suivant

2. Marges externes :

*Les **marges externes** d'un bloc sont situées au-delà de ses bordures. Elles servent à espacer les blocs entre eux. Elles sont définies sur chacun des côtés à l'aide des propriétés **margin-top** en haut, **margin-right** à droite, **margin-bottom** en bas et **margin-left** à gauche, ou globalement par la propriété raccourcie margin.*

Exemple:

```
<!DOCTYPE html>
<html>
   <head>
         <title>modèle des boîtes</title>
         <meta charset= "utf-8">
         <link rel = "stylesheet" href="style.css" />
   </head>

   <body>
      <div>
         <h1>les marges</h1>
         <p>bonjour UT</p>
          <p class="p1">je suis étudiant de la filière MIO</p>
         </div>
   </body>
</html>
```

code css

```
div{
background-color: #269;
width: 400px;
```

```
border: 5px ridge #444;
border-radius: 15px;
margin-top: 50px;
margin-left: 50px;
}
p {
    width: 300px;
    border: 3px solid yellow;
}
.p1{
 width: 300px;
 border: 10px solid red;
padding: 25px;
margin-left: 90px; /* (on laisse une marge de 90 à gauche)* /
}
```

9. Les ombrages en CSS

La propriété ***text-shadow*** permet de créer des ombres sur un texte.

Cette propriété s'utilise de la façon suivante :

text-shadow: 2px 2px 2px black;

Pour créer des ombres autour des boites à l'intérieur ou à l'extérieur nous pouvons utiliser la propriété ***boxshadow*** qui prend quatre valeurs différentes dans l'ordre suivant :

1. Le déplacement horizontal (vers la droite ou la gauche) de l'ombre ;
2. Le déplacement vertical (vers le bas ou le haut) de l'ombre ;
3. Le rayon de propagation de l'ombre ;
4. La couleur de l'ombre.

NB :

Deux types d'ombres existent en CSS :

- les `box-shadow`, les ombres ajoutées aux éléments de type bloc,
- les `text-shadow`, les ombres ajoutées au contenu textuel.

Exemple:

```
<!DOCTYPE html>
<html>
<head>
<title>modèle des boîtes</title>
<meta charset= "utf-8">
</head>

<body>
<div>
<h1>les marges</h1>
<p>bonjour UVS</p>
<p class="p1">je suis étudiant de la filière MIC</p>
</div>
</body> </html>
```

code css

```
div{
background-color: #269;
width: 400px; border:
5px ridge #444; border-
radius: 15px; margin-
top: 50px;
margin-left: 50px;
}

p {
width: 300px;
border: 3px solid yellow;
box-shadow: -5px -4px 5px yellow;
}
.p1{
width: 200px;
border: 10px solid red;
padding: 25px;
margin-left: 90px;
box-shadow: -10px 4px 5px lime inset;
}
```


les marges
bonjour UVS
je suis étudiant de la filière
MIC